

# Laboratoria nr 1 - wprowadzenie

dr inż. Joanna Kołodziejczyk

## 1 Zakres laboratoriów

1. Przeszukiwanie – algorytmy ślepe (dfs, bfs)
2. Przeszukiwanie – algorytmy z informacją (bfs – różne heurystyki)
3. Gry (algorytmy dla gier dwuosobowych) min-max, alpha-beta
4. Naiwny klasyfikator Bayesa wersja dyskretna i ciągła
5. Przeszukiwanie lokalne (stochastyczne) – hill-climbing, algorytm ewolucyjnym
6. Sieci neuronowe perceptron prosty
7. Sieci neuronowe MLP
8. Programowanie deklaratywne – Prolog

## 2 Ocena z laboratoriów

- Sprawdziany
  - zapowiedziane - zakres zawsze podany z wyprzedzeniem
  - początek zajęć około 5 min
  - na papierze
  - wyniki na MS Teams
- Prace programistyczne lub sprawozdania.
  - Pierwszy termin wykonania zadania na ocenę max 5.0
  - Drugi termin na ocenę max 4.0
  - Po terminie - nie sprawdzam
  - Program musi być omówiony, zaliczony

Ocena końcowa:

$$Ocena = 0.6 \cdot MeanTask + 0.4 \cdot MeanTest,$$

gdzie *MeanTask* to średnia ocena za zadania, a *MeanTest* to średnia ocena za sprawdziany

Ocena	Punkty za ocenę
2	2
2+	2.5
3-	2.75
3	3
3+	3.5
4-	3.75
4	4
4+	4.5
5-	4.75
5	5

Tabela 1: Skala ocen

### 3 Zadania programistyczne

- Dowolny wybrany przez studenta język programowania, każdy indywidualnie.
- Można zmienić język programowania w czasie semestru.
- Kod sprawdzany za pomocą systemu antyplagiatowego.
- Oceny wystawiane najwcześniej po 'Drugim terminie'.
- Źródła wysyłane przez MS Teams lub link do repozytorium (do uzgodnienia).
- Paczki tylko w formacie zip lub gzip (nie rar).
- Korzystanie z generatywnych AI zostaną omówiony przy każdym zadaniu indywidualnie.

### 4 Źródła:

Podstawowe zasoby - źródła wiedzy:

1. <http://wikizmsi.zut.edu.pl/wiki/SI>
2. Skrypt: [http://wikizmsi.zut.edu.pl/uploads/9/9e/Si\\_skrypt\\_draft.pdf](http://wikizmsi.zut.edu.pl/uploads/9/9e/Si_skrypt_draft.pdf)

### 5 Na dzisiejszych zajęciach:

1. Przeczytać i rozumieć problem n-hetmanów, o którym można przeczytać: [https://pl.wikipedia.org/wiki/Problem\\_o\\_ÅŻmiu\\_hetman\\_ÅŞw](https://pl.wikipedia.org/wiki/Problem_o_ÅŻmiu_hetman_ÅŞw)
2. Rozwiąż na kartce problem 4-hetmanów. Znajdź oba rozwiązania.

3. Zaproponuj strykturę/y danych do reprezentowania problemu  $n$ -hetmanów. Taka struktura musi poprawnie odwzorowywać: pustą szachownicę, jednego hetmana na dowolnym polu, dwóch itd. aż do  $n$ . Pomyśl o tablicach i wektorach.
4. Jak sprawdzać, czy hetmani się atakują? Napisz algorytm w postaci pseudokodu.

## 6 Na następne zajęcia:

1. Przeczytać i zrozumieć algorytmy przeszukiwania wszerz i w głąb. Rozdział 2 i 2.1.a ze skryptu.
2. Przygotuj się do sprawdzianu z zakresu działania tych algorytmów.